2015

Comparing the basins of attraction for Kanwar-Bhatia-Kansal family to the best fourth order method

Chun, Changbum; Neta, Beny

Elsevier

C. Chun, B. Neta, "Comparing the basins of attraction for Kanwar-Bhatia-Kansal family to the best fourth order method," Applied Mathematics and Computation, v. 296 (2015), pp. 277-292.

http://hdl.handle.net/10945/55177

Applied Mathematics and Computation 266 (2015) 277–292

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Comparing the basins of attraction for Kanwar–Bhatia–Kansal family to the best fourth order method

Changbum Chun [a,1], Beny Neta [b,*]

[a] Department of Mathematics, Sungkyunkwan University Suwon 440-746, Republic of Korea
[b] Department of Applied Mathematics Naval Postgraduate School Monterey, CA 93943, USA

ARTICLE INFO

MSC:
65H05
65B99

Keywords:
Iterative methods
Order of convergence
Multiple roots
Basin of attraction

ABSTRACT

There are relatively few optimal fourth order methods for solving nonlinear algebraic equations having roots of known multiplicity m. In a previous paper we have compared 5 such methods, two of which require the evaluation of the $(m-1)$st root. We have used the basin of attraction idea to recommend the best optimal fourth order method. Here we compare the family of methods developed by Kanwar et al. (2013)– to the best known method. We will also point out some mistake in deriving that family.

Published by Elsevier Inc.

1. Introduction

The field of numerical solution of nonlinear equations and nonlinear systems is vast and expanding. See, for example, Ostrowski [1], Traub [2], Neta [3] and the recent book by Petković et al. [4] and references therein. Zero finders of a scalar function f required in many branches of engineering sciences, physics, computer science, finance, to mention only a few. Most of the algorithms are for finding a simple root of a nonlinear equation $f(x)=0$, i.e. for a root α we have $f(\alpha)=0$ and $f'(\alpha)\neq 0$. In this paper we are interested in the case that α is a root of multiplicity $m>1$. Clearly, one can use the quotient $f(x)/f'(x)$ which has a simple root where $f(x)$ has a multiple root. Such an idea will not require a knowledge of the multiplicity, but on the other hand will require higher derivatives. Therefore the amount of information required to achieve a certain order of convergence is higher.

There are relatively very few methods for multiple roots when the multiplicity is known, see e.g. a method of order 1.5 [5], a method of order 2 [6], third order methods [7–17], and [18]. The fourth order methods [19–23] and [24]. Some of these methods are considered optimal in the sense of Kung and Traub [25], i.e. they have a maximal order of 2^n when using $n+1$ function- (and derivative-) evaluation per iteration step.

Here we concentrate on the optimal fourth-order family of methods [23]. We correct the errors in the paper and compare two members of the family to the best known fourth-order methods found in the literature.

The fourth-order family of methods [23] is given by

$$
\begin{aligned}
y_n &= x_n - \frac{2m}{m+2}\frac{f(x_n)}{f'(x_n)-pf(x_n)},\\
x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)-pf(x_n)}\,Q\!\left(\frac{f'(y_n)+hf'(x_n)}{tf'(x_n)-pf(x_n)}\right),
\end{aligned}
\tag{1}
$$

* Corresponding author. Tel.: 831 656 2235; fax: 831 656 2355.
E-mail addresses: cbchun@skku.edu (C. Chun), bneta@nps.edu, byneta@gmail.com (B. Neta).
1 Tel.: +82 31 299 4523; fax: +82 31 290 7033.

http://dx.doi.org/10.1016/j.amc.2015.05.069
0096-3003/Published by Elsevier Inc.

where Q is a real valued weight function satisfying

$$\begin{aligned}
&Q(\mu) = m,\\
&Q'(\mu) = -\frac{m^3\left(\frac{m}{m+2}\right)^{-m}}{4(1+m)},\\
&Q''(\mu) = \frac{m^4\left(\frac{m}{m+2}\right)^{-2m}}{4(m+1)^2},\\
&|Q'''(\mu)| < \infty,
\end{aligned} \tag{2}$$

and

$$\begin{aligned}
\mu &= \frac{2(m+1)}{m+2}\left(\frac{m}{m+2}\right)^{m-1} = \frac{2(m+1)}{m}\left(\frac{m}{m+2}\right)^{m},\\
t &= \frac{1}{m+1},\\
h &= -\left(\frac{m}{m+2}\right)^{m}.
\end{aligned} \tag{3}$$

Remark: The authors gave an erroneous value of μ which is corrected here.

The authors considered three members of the family. The third one choosing a quadratic polynomial for Q will not be considered here, since Chun and Neta [26] have shown that such a choice will give inferior results.

- KBK1

$$Q(x) = \frac{A}{x} + B, \tag{4}$$

where

$$\begin{aligned}
A &= m(1+m)\left(\frac{m}{m+2}\right)^{m},\\
B &= -\frac{m(m-2)}{2}.
\end{aligned} \tag{5}$$

- KBK2

$$Q(x) = \frac{A}{(x+C)^2} + B, \tag{6}$$

where

$$\begin{aligned}
A &= \frac{27}{8}(m+1)^2\left(\frac{m}{m+2}\right)^{2m},\\
B &= -\frac{3}{8}m^2 + m,\\
C &= \frac{m+1}{m}\left(\frac{m}{m+2}\right)^{m}.
\end{aligned} \tag{7}$$

Remark: Actually Kanwar et al. [23] have suggested a linear polynomial in the denominator, but this will not be different from the first case. We have decided here to try something close to what they have, namely a quadratic in the denominator.
The methods we compare to these two are LCN6 and ZCS3 given below. These methods were shown to perform better than other fourth order methods for multiple roots [42]. We also use ZCS1 and ZCS5 below, which were the best options suggested in [26].

- LCN6 [21]

$$\begin{aligned}
y_n &= x_n - \frac{2m}{m+2}\frac{f(x_n)}{f'(x_n)},\\
x_{n+1} &= x_n - a_3\frac{f(x_n)}{f'(x_n)} - \frac{f(x_n)}{b_1 f'(x_n) + b_2 f'(y_n)},
\end{aligned} \tag{8}$$

where

$$a_3 = -\frac{1}{2}m(m-2), \quad b_1 = -\frac{1}{m}, \quad b_2 = \frac{1}{m\left(\frac{m}{m+2}\right)^m}.$$

For the method presented by Zhou et al. [24]

$$y_n = x_n - \frac{2m}{m+2}\frac{f(x_n)}{f'(x_n)},$$
$$x_{n+1} = x_n - \phi(t_n)\frac{f(x_n)}{f'(x_n)}, \tag{9}$$

where $t_n = \dfrac{f'(y_n)}{f'(x_n)}$ and ϕ is at least twice differentiable function satisfying the following conditions:

$$\phi(\lambda) = m,$$
$$\phi'(\lambda) = -\frac{1}{4}m^3\left(\frac{m+2}{m}\right)^m,$$
$$\phi''(\lambda) = \frac{1}{4}m^4\left(\frac{m+2}{m}\right)^{2m}, \tag{10}$$

and $\lambda = \left(\frac{m}{m+2}\right)^{m-1}$, we will consider the following functions:

- ZCS3 [24]

$$\phi(x) = \frac{B+Cx}{1+Ax}, \tag{11}$$

where $A = -\left(\frac{m+2}{m}\right)^m, B = -\frac{m^2}{2}, C = \frac{1}{2}m(m-2)\left(\frac{m+2}{m}\right)^m$.
Note that $A = b_2/b_1,\ B = a_3 + 1/b_1,\ C = a_3 b_2/b_2$ and thus LCN6 is identical to ZCS3. The results for ZCS3 will not appear here.

- ZCS1 [26]

$$\phi(x) = \frac{b+cx+dx^2}{1+ax+gx^2}. \tag{12}$$

where

$$b = \frac{m}{8}\left((m+2)^2\lambda m a + (m+2)\lambda^2 m^2 g + m^3 + 6m^2 + 8m + 8\right),$$
$$c = -\frac{m}{4\lambda}\left((m^3+3m^2+2m-4)\lambda a + (m^2+m-2)\lambda^2 m g + m(m+2)(m+3)\right),$$
$$d = \frac{m}{8\lambda^2}\left(m^2(m+2)\lambda a + (m^3-4m+8)\lambda^2 g + m(m+2)^2\right), \tag{13}$$

with $a = -4, g = 0$.

- ZCS5 [26]
Same weight function ϕ (12) with $a = -6.01, g = 8.04$.

The idea of using basins of attraction was initiated by Stewart [27] and followed by the works of Amat et al. [28–30], and [31], Scott et al. [32], Chun et al. [33,40], Magreñán [34], Argyros et al. [35], Chicharro et al. [37], and Cordero et al. [36,38]. The only papers comparing basins of attraction for methods to obtain multiple roots are due to Neta et al. [41], Neta and Chun [18], [42], and Chun and Neta [26].

This comparative study is numerical, but in order to try and understand why certain methods perform better than others we are going to find the extraneous fixed points (see Vrcsay and Gilbert [43]). In the next section, we find the extraneous fixed points of the new methods for various values of the multiplicity. In Section 3 we will experiment with the 5 methods discussed above to solve 6 different equations having roots with various multiplicities. We close with concluding remarks on best performer among the 5 methods.

2. Extraneous fixed points

Iterative solution of a nonlinear equation is the same as finding a fixed point of an iteration function. Many iteration functions have fixed points that are not zeros of the function of interest. Those points are called extraneous fixed points (see Vrcsay and Gilbert [43]). The extraneous fixed points could be attractive, repulsive or indifferent. In the case of an attractive point, the iteration sequence will be trapped and give erroneous results. Even if the extraneous fixed points are repulsive or indifferent the iteration sequence may converge to a root not close to the initial guess. As we can see in the next section, the new methods have both attractive and repulsive extraneous fixed points. We should also mention here that we have found previously [39] that if the extraneous fixed points are on the imaginary axis the method performs better.

The methods discussed here can be written as

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}H_f(x_n).$$

Table 1
$H_f(x_n)$ for our fourth order methods for multiple roots.

Method	H_f
KBK1	$\frac{f'(x_n)}{f'(x_n)-pf(x_n)}Q\left(\frac{f'(y_n)+hf'(x_n)}{tf'(x_n)-pf(x_n)}\right)$
KBK2	$\frac{f'(x_n)}{f'(x_n)-pf(x_n)}Q\left(\frac{f'(y_n)+hf'(x_n)}{tf'(x_n)-pf(x_n)}\right)$
LCN6	$a_3+\frac{1}{b_1+b_2\frac{f'(y_n)}{f'(x_n)}}$
ZCS1	$\frac{b+c\frac{f'(y_n)}{f'(x_n)}+d\left(\frac{f'(y_n)}{f'(x_n)}\right)^2}{1+a\frac{f'(y_n)}{f'(x_n)}+g\left(\frac{f'(y_n)}{f'(x_n)}\right)^2}$
ZCS5	$\frac{b+c\frac{f'(y_n)}{f'(x_n)}+d\left(\frac{f'(y_n)}{f'(x_n)}\right)^2}{1+a\frac{f'(y_n)}{f'(x_n)}+g\left(\frac{f'(y_n)}{f'(x_n)}\right)^2}$

Clearly the root α of $f(x)$ is a fixed point of the method. The points $\xi \neq \alpha$ at which $H_f(\xi) = 0$ are also fixed points of the family, since the second term on the right vanishes.

It is easy to see that $H_f(x_n)$ for our methods is given in Table 1.

Theorem 1. *The extraneous fixed points for KBK1 can be found by solving*

$$\frac{f'(\xi)}{f'(\xi)-pf(\xi)}Q\left(\frac{f'(y(\xi))+hf'(\xi)}{tf'(\xi)-pf(\xi)}\right)=0, \tag{14}$$

where $Q(x)$ is given by (4).

Proof. The extraneous fixed points can be found by solving (14). For the polynomial $(z^2-1)^m$ this leads to a very complicated equation. We will give the solution for several values of the multiplicity. All the computations were done using Maple.

For $m = 2$ then (14) becomes

$$\frac{N_{21}(z)}{D_{21}(z)}=0 \tag{15}$$

where

$$N_{21}(z)=2(z^2-4z-1)^2(3z^2-4pz-3)$$

$$D_{21}(z)=z^6-12pz^5+45z^4-40pz^3-13z^2-76pz-33.$$

The roots are as follows:

$$\xi=2p\pm\sqrt{5},\ \xi=\frac{2}{3}p\pm\frac{\sqrt{13}}{3}.$$

The fixed points $\xi=\frac{2}{3}p\pm\frac{\sqrt{13}}{3}$ are attractive and $\xi=2p\pm\sqrt{5}$ are repulsive.

If $m = 3$ then (14) becomes

$$\frac{N_{31}(z)}{D_{31}(z)}=0, \tag{16}$$

where

$$\begin{aligned}N_{31}(z) &= 9z(300z^{14}-9675pz^{13}+126150z^{12}-832950pz^{11}+2767050z^{10}\\&\quad-3287925pz^9-3685442z^8+8247900pz^7+3545768z^6-2168325pz^5\\&\quad-4107702z^4-1952550pz^3-265582z^2+3525pz-542)\\D_{31}(z) &= (z^2-6pz-1)(225z^{13}-6750pz^{12}+79650z^{11}-452250pz^{10}\\&\quad+1137375z^9-358858pz^8-2421300z^7+498532pz^6+17775z^5\\&\quad+1792902pz^4+1199250z^3+146182pz^2-12975z+242p).\end{aligned}$$

The roots are as follows:

$\xi=-0.284894248793855p\pm 0.494927802751884i,$
$\xi=0.0116810658529917p\pm 0.0404319234289949i,$

$\xi = 5.22268367454299p \pm 0.721236947288976i,$
$\xi = 6.55813812917723p \pm 1.04032583323227i,$
$\xi = -1.10603157199272p, \xi = -0.341152346885745p,$
$\xi = -0.314487294122869p, \xi = 7.24009285845507p,$
$\xi = 1.13181190403280p, \xi = 2.62454920895449p,$
$\xi = 0.$

The fixed points $\xi = -1.10603157199272p, \xi = 1.13181190403280p, \xi = 2.62454920895449p$ are attractive and all the other are repulsive.

For $m = 4, 5$, the results are messy and we will only give the extraneous fixed points.

If $m = 4$ then

$\xi = -1.10447295044898p \pm 0.235948103022870i,$
$\xi = -0.263867071102825p \pm 0.0113610569179989i,$
$\xi = -0.193091272258953p \pm 0.549746273535783i,$
$\xi = 0.0411965767242920p \pm 0.0464573274186521i,$
$\xi = 1.08042622583025p \pm 0.280162912628156i,$
$\xi = 6.824984112p \pm 0.6446291884i,$
$\xi = 7.859627214p \pm 1.391963893i,$
$\xi = 9.007938618p \pm 1.046117299i,$
$\xi = 9.47575305714824p, \xi = -0.280377296299590p,$
$\xi = 0.0123324117643947p, \xi = 3.48680892222868p,$
$\xi = 0.$

The fixed point $\xi = 3.48680892222868p$ is attractive. All the other are repulsive.

If $m = 5$ then

$\xi = -0.991172725906456p \pm 0.365922951482879i,$
$\xi = -0.234907875272102p \pm 0.0169053658357632i,$
$\xi = -0.136574281119478p \pm 0.538380927782273i,$
$\xi = 0.0190608974626157p \pm 0.0146507196899269i,$
$\xi = 0.0544823598775080p \pm 0.0475957741655994i,$
$\xi = 0.940006120815937p \pm 0.386558866911588i,$
$\xi = 8.617693331p \pm 0.5504966363i,$
$\xi = 9.345936186p \pm 1.374574086i,$
$\xi = 10.40706818p \pm 1.505968769i,$
$\xi = 11.2739519p0 \pm 0.9516092011i,$
$\xi = -0.227559960470804p, \xi = -0.218397165171802p,$
$\xi = 1.29832298218231p, \xi = 4.39423743723399p,$
$\xi = 11.6018727959477p, \xi = -1.27289760612121p,$
$\xi = 0.$

All the fixed points are repulsive. □

Theorem 2. *The extraneous fixed points for KBK2 can be found by solving*

$$\frac{f'(\xi)}{f'(\xi) - pf(\xi)} Q\left(\frac{f'(y(\xi)) + hf'(\xi)}{tf'(\xi) - pf(\xi)}\right) = 0, \tag{17}$$

where $Q(x)$ is given by (6).

Proof. We will give the solution for several values of the multiplicity. All the computations were done using Maple.

For $m = 2$ then (17) becomes

$$8z\frac{N_{21}(z)}{D_{21}(z)} = 0, \tag{18}$$

where

$$\begin{aligned} N_{21}(z) = {} & 63z^{16} - 1668pz^{15} + 18472z^{14} - 109092pz^{13} + 356388z^{12} - 560276pz^{11} \\ & + 48920z^{10} + 930572pz^9 - 403782z^8 - 1002380pz^7 + 155672z^6 + 585492pz^5 \\ & + 384036z^4 + 156068pz^3 + 29992z^2 + 1284pz + 63 \\ D_{21}(z) = {} & (3z^{10} - 44pz^9 + 209z^8 - 208pz^7 - 898z^6 + 1784pz^5 - 638z^4 + 1328pz^3 \\ & + 1327z^2 + 212z - 3)(3z^8 - 32pz^7 + 84z^6 + 96pz^5 - 430z^4 + 160pz^3 - 428z^2 \\ & - 224pz + 3). \end{aligned}$$

The roots are as follows:

$\xi = -0.737290308820108p \pm 0.319624594752665i$,
$\xi = -0.4068010585p \pm 0.006967763710i$,
$\xi = -0.103289124467843p \pm 0.402904439859927i$,
$\xi = -0.0192422953705217p \pm 0.0479303677745048i$,
$\xi = 1.825321759p \pm 0.5329670349i$,
$\xi = 3.414458221p \pm 0.3397423568i$,
$\xi = 4.387833851p \pm 0.7806655553i$,
$\xi = 4.877104194p \pm 0.4312095366i$,
$\xi = 0$.

All the fixed points are repulsive.

For $m = 3, 4, 5$ the results are messy and we will only give the extraneous fixed points.

If $m = 3$ then

$\xi = -0.288870919844895p \pm 0.451851674787056i$,
$\xi = 0.00730540713755696p \pm 0.0435709612369002i$,
$\xi = 5.286024756p \pm 0.6666643783i$,
$\xi = 6.528420735p \pm 0.9772247737i$,
$\xi = -0.906229488870198p \pm 0.229228180242995i$,
$\xi = -0.317325904063935p \pm 0.0291540211560247i$,
$\xi = 5.886437510p \pm 0.9426005995i$,
$\xi = 6.935623938p \pm 0.5383183692i$,
$\xi = 0.102079235788502p \pm 0.208820131572240i$,
$\xi = -1.09077147022139p, \xi = -0.347546441909348p$,
$\xi = -0.310878634469743p, \xi = 7.17509026881993p$,
$\xi = 1.11141123809958p, \xi = 2.45943508145007p$,
$\xi = 0.773997625223165p, \xi = 1.71157388885550p$,
$\xi = 5.17239204026288p, \xi = -0.00913413438002196p$,
$\xi = 0$.

The fixed points $\xi = -1.09077147022139p$, $\xi = 1.11141123809958p$, $\xi = -0.288870919844895p \pm 0.451851674787056i$, $\xi = 2.45943508145007p$, $\xi = -0.906229488870198p \pm 0.229228180242995i$, $\xi = 0.773997625223165p$ are attractive, and all the other are repulsive.

If $m = 4$ then

$\xi = -1.102995613p \pm 0.2338177764i$,
$\xi = -0.8776703279p \pm 0.4651129818i$,
$\xi = -0.2746851632p \pm 0.03629129494i$,
$\xi = -0.2629611081p \pm 0.01197124888i$,
$\xi = -0.1950174361p \pm 0.5445834540i$,
$\xi = 0.01489225777p \pm 0.02384645642i$,
$\xi = 0.04086014961p \pm 0.04708420948i$,
$\xi = 0.3012151476p \pm 0.4421499212i$,
$\xi = 1.078558879p \pm 0.2778593119i$,
$\xi = 1.371282915p \pm 0.2605921345i$,
$\xi = 6.837624895p \pm 0.6373958788i$,
$\xi = 7.349141662p \pm 0.9910448387i$,
$\xi = 7.860583618p \pm 1.379163813i$,
$\xi = 8.424307293p \pm 1.198697637i$,
$\xi = 8.999935474p \pm 1.037841929i$,
$\xi = 9.222463893p \pm 0.5155935874i$,
$\xi = -0.2824291770p, \xi = -1.291008389p$,
$\xi = 0.01201026752p, \xi = -0.2464776425p$,
$\xi = 3.417821714p, \xi = 0.1726298421p$,
$\xi = 9.464626229p, \xi = 6.870061775p$,
$\xi = 0$.

The fixed points $\xi = -1.291008389p, \xi = 3.417821714p$ are attractive. All the other are repulsive.

If $m = 5$ then

$\xi = -1.316146757p \pm 0.2369419842i$,
$\xi = -0.9910444248p \pm 0.3651650643i$,

$\xi = -0.8065668923p \pm 0.5380386645i,$
$\xi = -0.2440963282p \pm 0.04018633434i,$
$\xi = -0.234221586p5 \pm 0.01667715634i,$
$\xi = -0.2090308601p \pm 0.01346414878i,$
$\xi = -0.1372817413p \pm 0.5371114794i,$
$\xi = 0.01898257312p \pm 0.01468189060i,$
$\xi = 0.02707926663p \pm 0.03290816751i,$
$\xi = 0.05442466812p \pm 0.04781259321i,$
$\xi = 0.2286885479p \pm 0.5242720777i,$
$\xi = 0.9396284263p \pm 0.3857558847i,$
$\xi = 1.148102056p \pm 0.4438098408i,$
$\xi = 8.622143272p \pm 0.5485967097i,$
$\xi = 9.034099078p \pm 0.9002610406i,$
$\xi = 9.347654703p \pm 1.370446078i,$
$\xi = 9.875405491p \pm 1.370182461i,$
$\xi = 10.40587547p \pm 1.502084119i,$
$\xi = 10.80155473p \pm 1.178752509i,$
$\xi = 11.27085477p \pm 0.9493578233i,$
$\xi = 11.38050393p \pm 0.4581216198i,$
$\xi = -1.272342878p, \xi = 0.01447312847p,$
$\xi = -0.2297025620p, \xi = -0.2176911481p,$
$\xi = 0.1420120696p, \xi = 4.353455969p,$
$\xi = 1.297885597p, \xi = 1.417327128p,$
$\xi = 8.708041422p, \xi = 11.59811618p,$
$\xi = 0.$

All the fixed points are repulsive. □

Theorem 3. *The extraneous fixed points for LCN6 can be found by solving*

$$\frac{f'(y(\xi))}{f'(\xi)} = -\frac{1/a_3 + b_1}{b_2}. \tag{19}$$

If $a_3 = 0$, then there are no extraneous fixed points. This happens when $m = 2$. All the fixed points are repulsive for $m = 3, 4, 5$.

The proof can be found in [42].

Theorem 4. *The extraneous fixed points for ZCS1 and ZCS5 can be found by solving*

$$b + c\frac{f'(y(\xi))}{f'(\xi)} + d\left(\frac{f'(y(\xi))}{f'(\xi)}\right)^2 = 0. \tag{20}$$

The proof can be found in [26].

3. Numerical experiments

We experimented with the above 5 methods: KBK1, KBK2, LCN6, ZCS1, and ZCS5. We ran the 5 methods on 6 different polynomials having multiple roots with multiplicity $m = 2, 3, 4$ and 5. We have included the basins of attractions to show the best and worst of the 5 methods. In general we prefer to have a more qualitative comparison, by computing the average number of iterations required for convergence per initial point, the standard deviation and the CPU time required. In each case we have taken a 6×6 square centered at the origin. The total number of initial points in the square is 360,000 uniformly spaced. The code will assign a color to each point based on the root it converged to. If the method did not converge after 40 iterations the code will assign a black color to the point. We have also used the intensity of the color to indicate the number of iterations, i.e. the lighter the shade the faster the method converged to that root.

Example 1. In our first example, we have taken the polynomial

$$p_1(z) = (z^2 - 1)^2 \tag{21}$$

whose roots $z = \pm 1$ are both real and of multiplicity $m = 2$. The basins are plotted in Fig. 1. One should expect the domain to be divided to two by the imaginary axis. Any point to the left of this axis will give a sequence converging to $z = -1$ and any point to the right to $z = +1$. The top left sub plot shows the basins for KBK1 and the top right for KBK2. Notice the domain is not divided equally. Thus there are initial points closer to the root $z = -1$ that converge to $z = +1$. Note that there is a small region to the right of $z = +1$ where the iteration converges to the root $z = -1$. The other methods are more balanced but have black regions near the interface (the imaginary axis). Now we look at Table 2 to note that the new methods KBK1 and KBK2 converge faster on

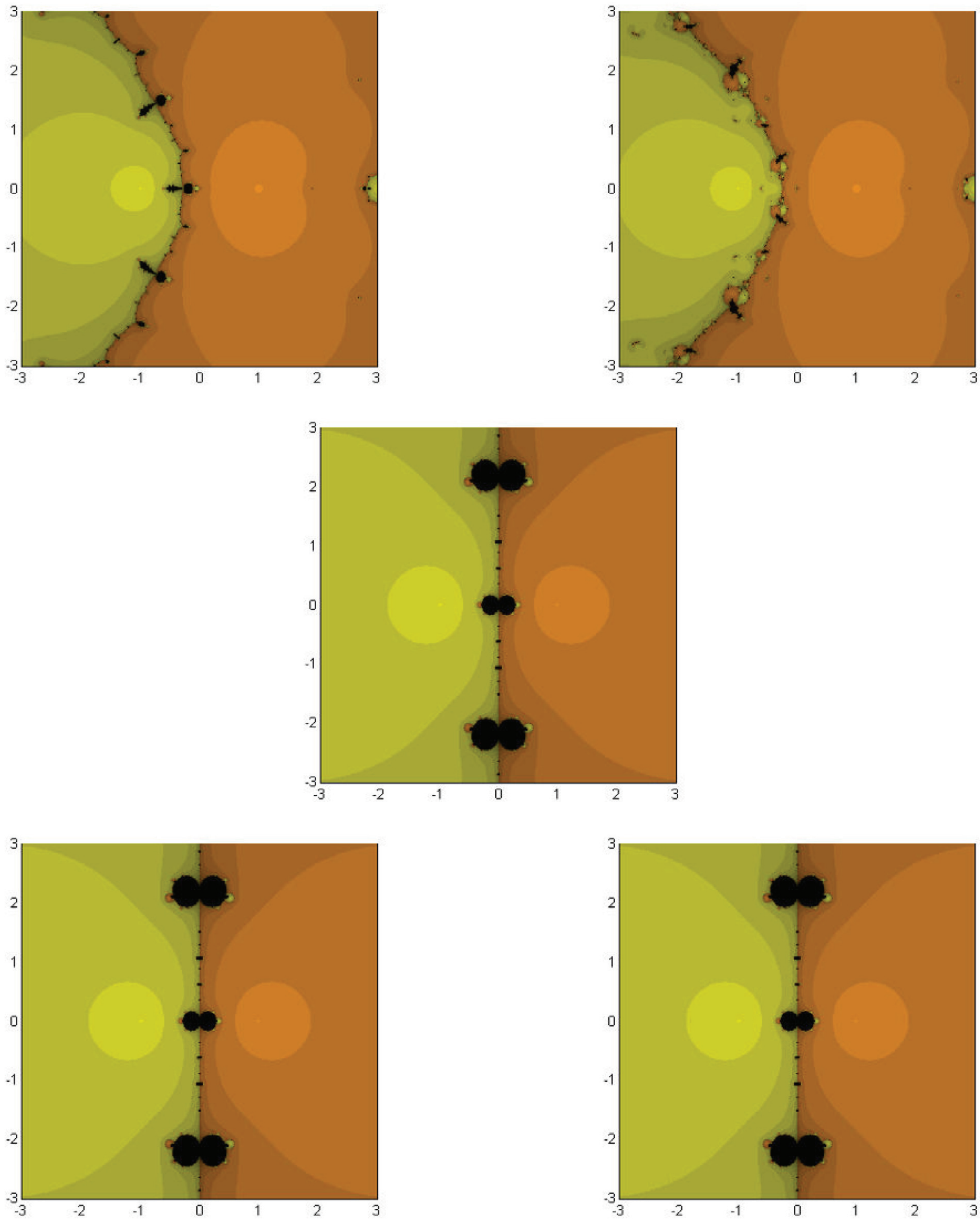

Fig. 1. The top left for KBK1, top right for KBK2, center for LCN6, the bottom left for ZCS1 and the bottom right for ZCS5 for the roots of the polynomial $(z^2-1)^2$.

Table 2
Average number of iterations per point for each example (1–6) and each of the 5 methods.

Example	KBK1	KBK2	LCN6	ZCS1	ZCS5
1	3.6874	3.6668	4.4206	4.4206	4.4208
2	4.2703	4.6396	4.4132	6.6396	6.6396
3	4.2750	4.4150	4.5919	4.7718	4.0634
4	5.4648	5.5707	6.1549	4.4840	4.0613
5	5.5498	7.0038	5.9339	19.9285	18.3734
6	4.9918	4.1575	6.6395	6.8108	6.3289
Average	4.7065	4.9089	5.3590	7.8425	7.3146

Table 3
Standard deviation for each example (1–6) and each of the 5 methods.

Example	KBK1	KBK2	LCN6	ZCS1	ZCS5
1	2.8247	2.4228	6.1727	6.1727	6.1727
2	2.3430	3.7327	1.9011	9.5553	9.5553
3	1.7744	2.8354	2.0825	4.4421	1.5483
4	2.7034	3.3154	4.0627	2.7098	1.4917
5	2.6551	6.6416	3.5449	14.0504	12.2306
6	6.5085	3.2381	9.5554	4.4168	1.5400

Table 4
CPU time (in seconds) required for each example (1–6) and each of the 5 methods using a Dell Multiplex-900.

Example	KBK1	KBK2	LCN6	ZCS1	ZCS5
1	332.85	326.51	256.34	233.56	246.48
2	699.29	761.01	476.44	457.41	471.93
3	596.26	631.60	437.44	415.74	358.41
4	884.46	902.59	635.42	890.64	741.11
5	906.74	1137.38	656.38	1637.90	1535.47
6	582.60	498.62	517.90	471.12	458.83
Average	667.03	709.62	496.65	684.39	635.37

average, also the standard deviation (Table 3) is smaller. On the other hand these methods require more CPU time (in Seconds) to run the example on all 360,000 initial points (see Table 4). We can summarize that the methods KBK1 and KBK2 are more computationally expensive and do not always converge to the closest root.

Example 2. The second example is a polynomial whose roots are all of multiplicity three. The roots are $-2.68261500670705 \pm .358259359924043i$, 1.36523001341410, i.e.

$$p_2(z) = (z^3 + 4z^2 - 10)^3. \tag{22}$$

The basins are displayed in Fig. 2. Here we see that KBK1 and KBK2 have smaller basins for the complex roots. The division is slightly better for the other 3 methods. The methods ZCS1 and ZCS5 have more black points than the other schemes (see the bottom sub plots of Fig. 2). This feature shows in Table 2 where the largest average number of iterations per point is highest. Also the spread is highest (see Table 3). Again we see that KBK1 and KBK2 require the most CPU time (see Table 4).

Example 3. The third example is a polynomial whose roots are all of multiplicity four. The roots are the three roots of unity, i.e.

$$p_3(z) = (z^3 - 1)^4. \tag{23}$$

The plots of the basins are given in Fig. 3. It is clear that the best one is ZCS5 (bottom right sub-plot). The highest average number of iterations per point (see Table 2) is for ZCS1 which has black points (bottom left sub-plot). The others are about the same, but as the plots show, the basin for $z = 1$ for KBK1 and KBK2 is larger than that for the other methods. LCN6 is the fastest followed by ZCS5. The slowest is KBK2 (see Table 4).

Example 4. In our next example we took the polynomial

$$p_4(z) = (z^4 - 1)^5 \tag{24}$$

where the roots (all of multiplicity $m = 5$) are symmetrically located on the axes. The plots of the basins are given in Fig. 4. Again the basins for KBK1 and KBK2 are not as symmetric as the basins for the other methods. LCN6 (center sub-plot) seem to perform best. This method requires the highest number of iterations per point (see Table 2) with the highest spread (see Table 3). On the other hand the CPU time was the smallest.

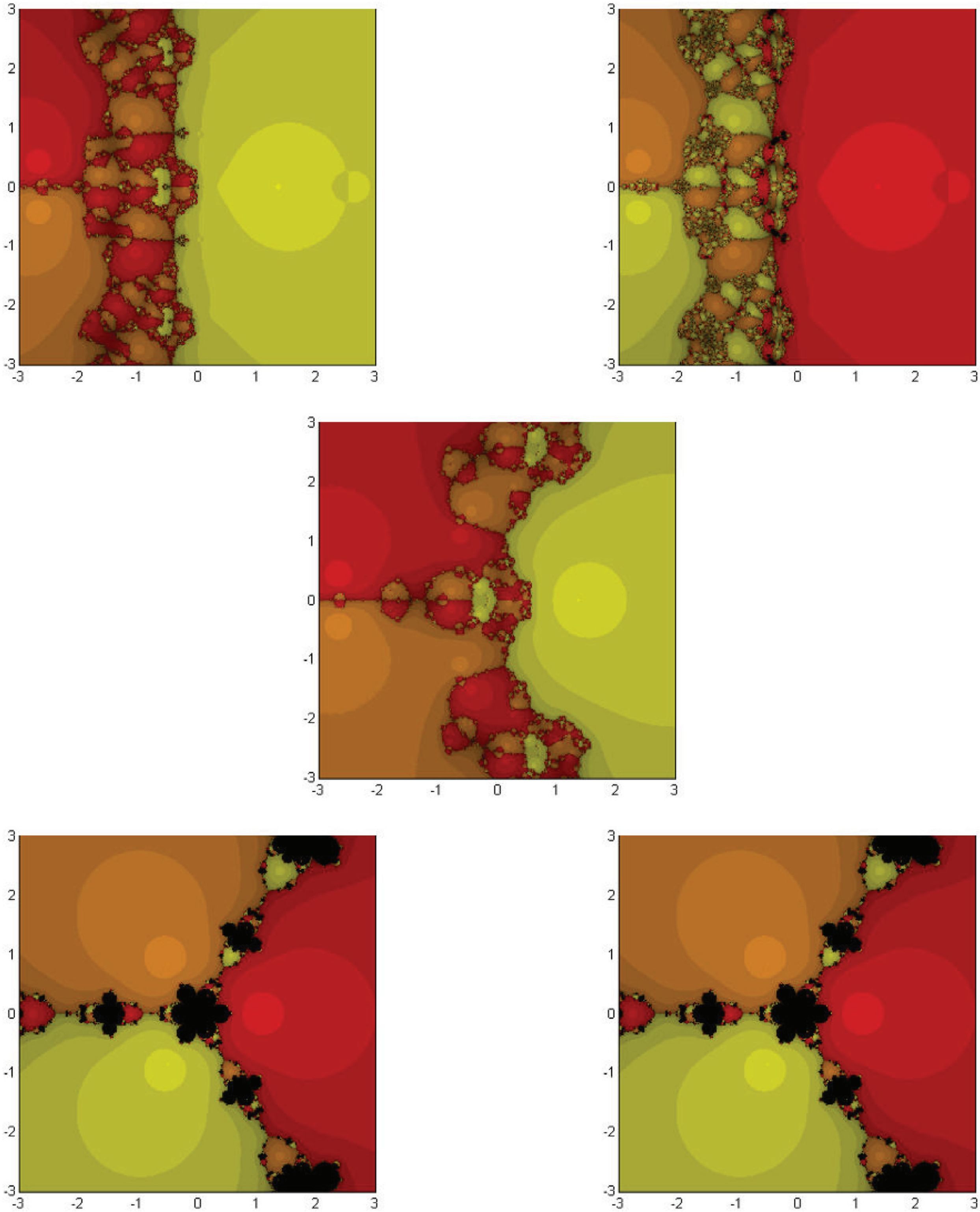

Fig. 2. The top left for KBK1, top right for KBK2, center for LCN6, the bottom left for ZCS1 and the bottom right for ZCS5 for the roots of the polynomial $(z^3 + 4z - 10)^3$.

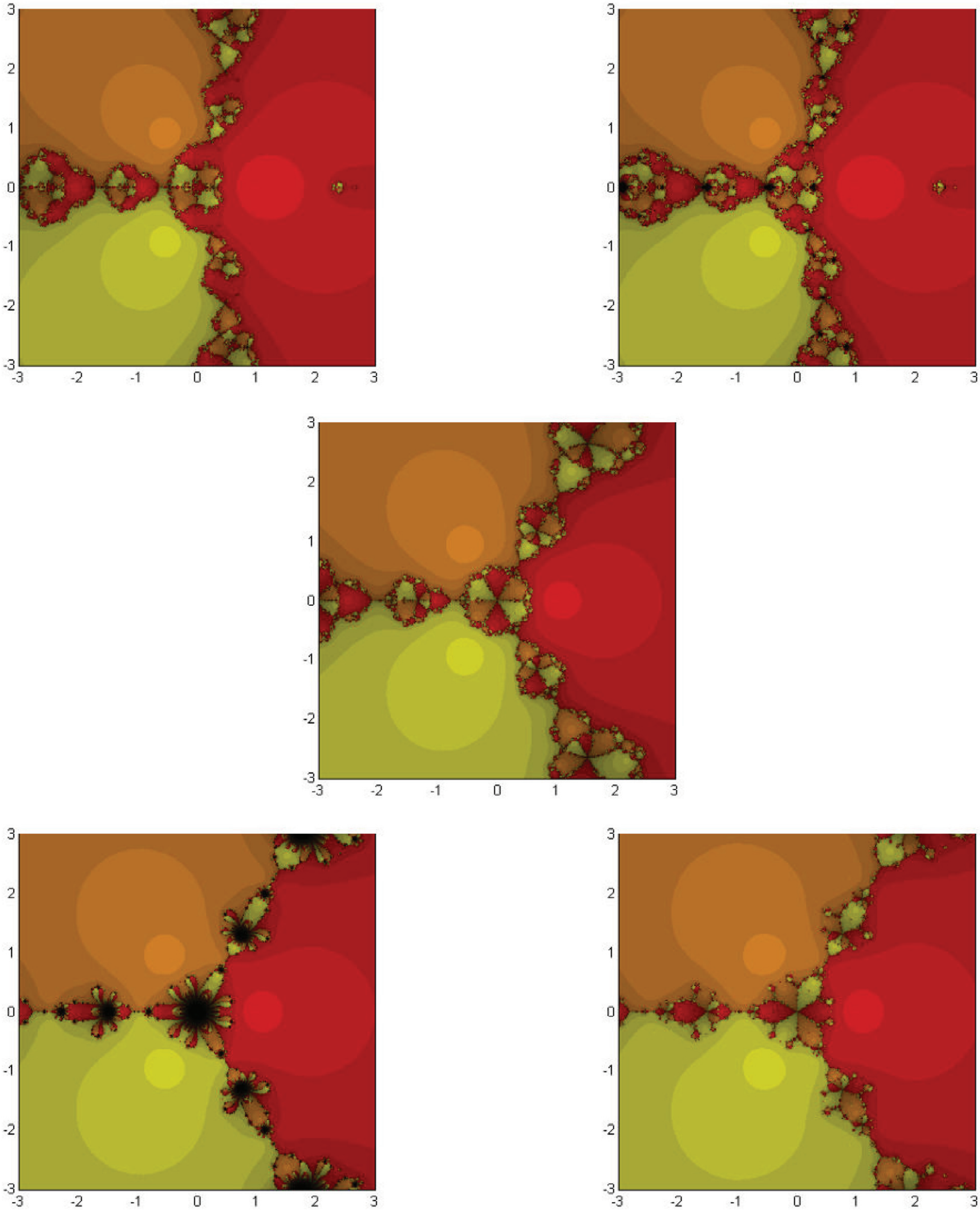

Fig. 3. The top left for KBK1, top right for KBK2, center for LCN6, the bottom left for ZCS1 and the bottom right for ZCS5 for the roots of the polynomial $(z^3-1)^4$.

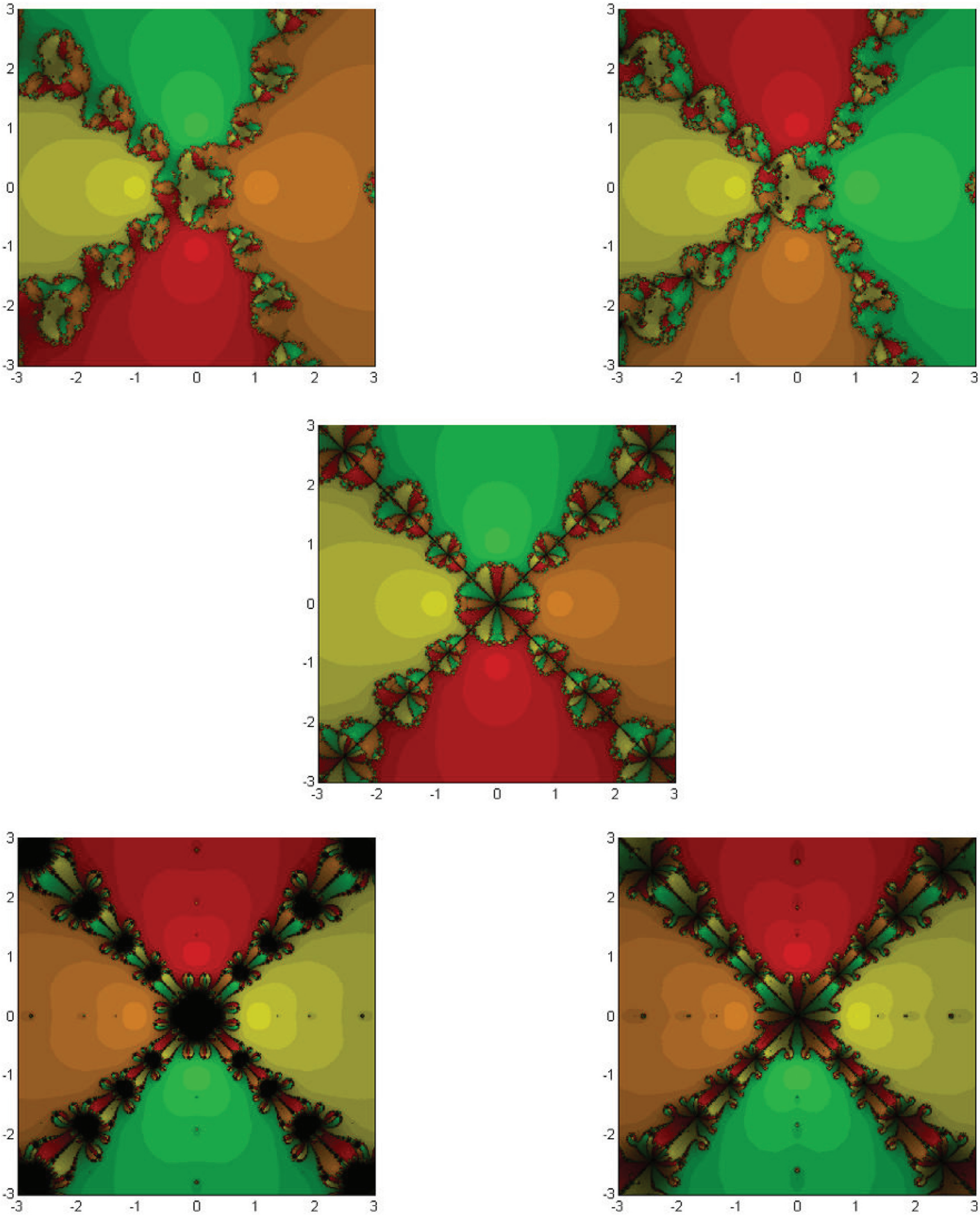

Fig. 4. The top left for KBK1, top right for KBK2, center for LCN6, the bottom left for ZCS1 and the bottom right for ZCS5 for the roots of the polynomial $(z^4 - 1)^5$.

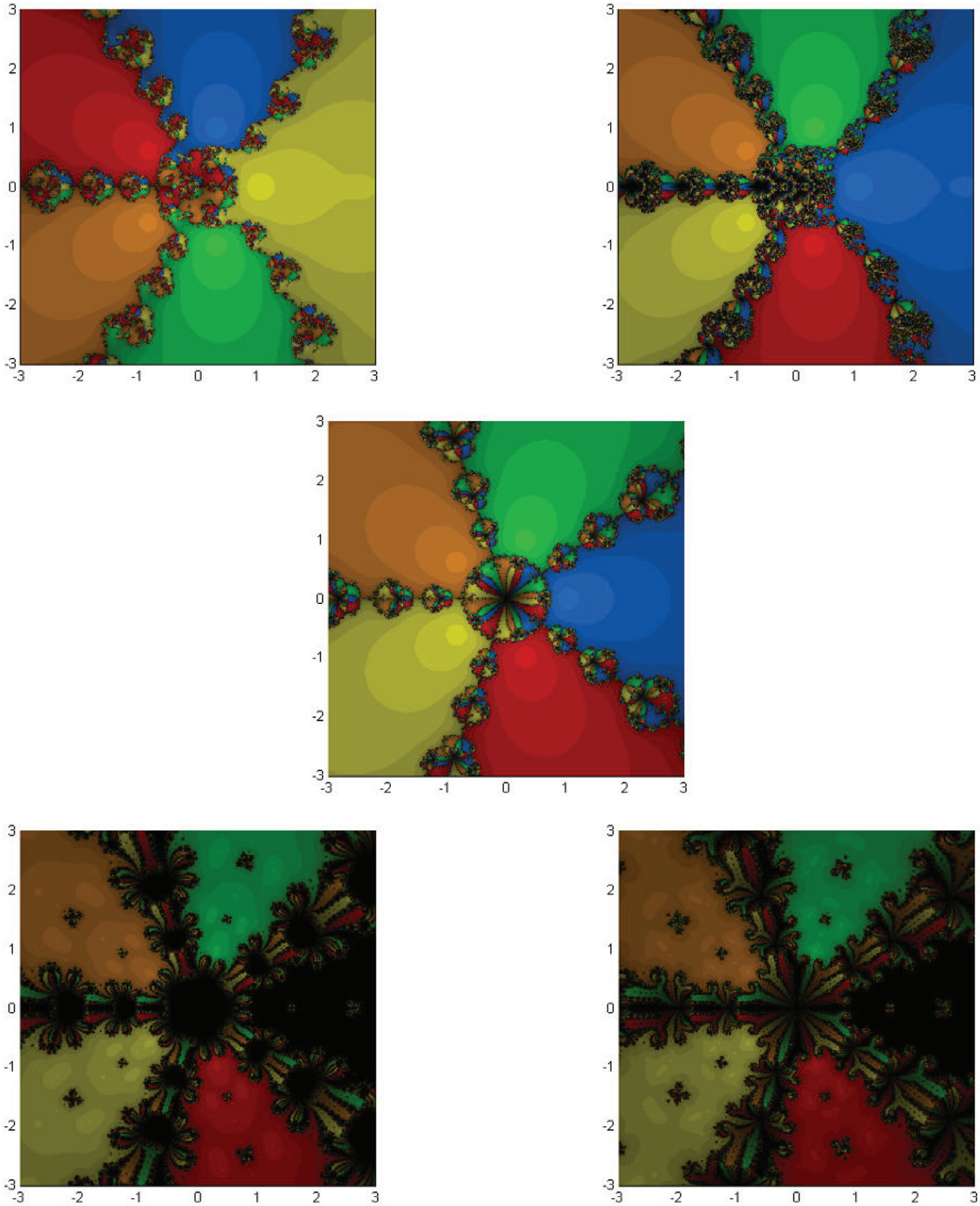

Fig. 5. The top left for KBK1, top right for KBK2, center for LCN6, the bottom left for ZCS1 and the bottom right for ZCS5 for the roots of the polynomial $(z^5 - 1)^3$.

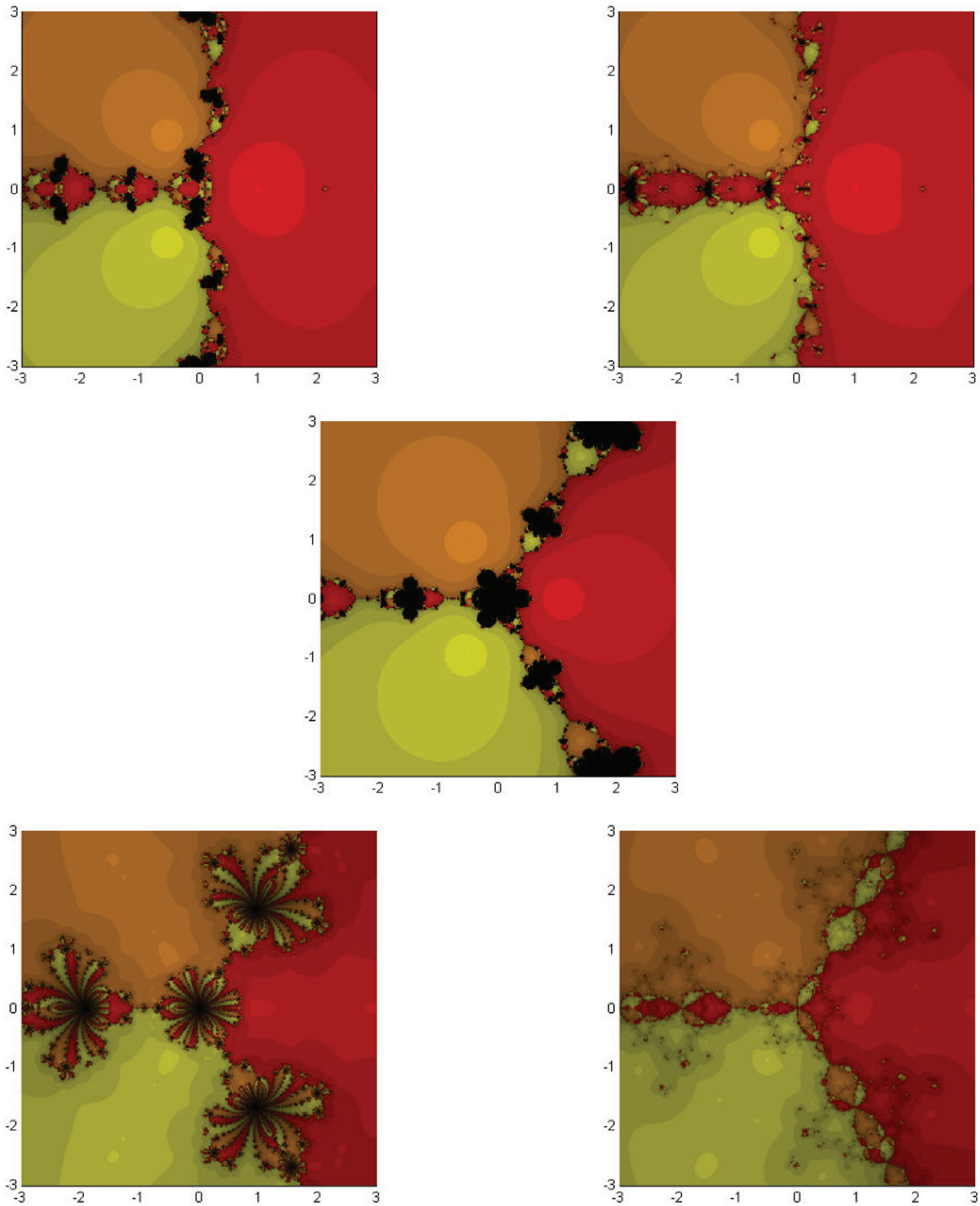

Fig. 6. The top left for KBK1, top right for KBK2, center for LCN6, the bottom left for ZCS1 and the bottom right for ZCS5 for the roots of the polynomial $(z^3-1)^2$.

Example 5. Our next example is having triple roots. The polynomial has the five roots of unity,

$$p_5(z) = (z^5 - 1)^3. \tag{25}$$

The basins are given in Fig. 5. The conclusions are the same as before. The average number of iterations per point for LCN6 is second lowest. Even though KBK1 has the lowest number, the basins are not equal in size as for LCN6. ZCS1 and ZCS5 have the highest average number of iterations per point and it shows in the plots (bottom row) as black points.

Example 6. In our last example we have the 3 roots of unity all with multiplicity two.

$$p_6(z) = (z^3 - 1)^2. \tag{26}$$

This is similar to example 3 except for the multiplicity. The plots of the basins are given in Fig. 6. In terms of the size of the basins, there is a slight difference in ZCS5 (bottom right) and ZCS1 (bottom left). There are more black points in LCN6 and KBK1. Therefore the methods perform differently when changing the multiplicity.

4. Conclusion

In order to come up with the best performer(s), we have averaged the values in Tables 2 and 4. On average LCN6 is the fastest. KBK1 came in fourth and KBK2 was last. In terms of the average number of iterations per point, LCN6 came third after KBK1 and KBK2. In spite of that we still recommend LCN6 over KBK1 and KBK2 because of the size of the basins, i.e. LCN6 iterates converge to the closest root. We note that LCN6 has no extraneous fixed points for $m = 2$ and those points for $m > 2$ are all repulsive. It seems that the fact that the extraneous fixed points for KBK1 and KBK2 depend on the parameter p that may change during the iteration caused the unbalance between the basins. The basin for the real positive root was larger than the other basins for the methods KBK1 and KBK2.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2013R1A1A2005012).

References

[1] A.M. Ostrowski, Solution of Equations in Euclidean and Banach Space, Academic Press, New York, 1973.
[2] J.F. Traub, Iterative Methods for the Solution of Equations, Chelsea Publishing Company, New York, 1977.
[3] B. Neta, Numerical Methods for the Solution of Equations, Net-A-Sof, California, 1983.
[4] M.S. Petković, B. Neta, L.D. Petković, J. Džunić, Multipoint Methods for Solving Nonlinear Equations, Elsevier, Waltham, MA, 2013.
[5] W. Werner, Iterationsverfahren höherer ordnung zur lösung nicht linearer gleichungen, Z. Angew. Math. Mech. 61 (1981) T322–T324.
[6] E. Schröder, über unendlich viele algorithmen zur auflösung der gleichungen, Math. Annal. 2 (1870) 317–365.
[7] C. Dong, A basic theorem of constructing an iterative formula of the higher order for computing multiple roots of an equation, Math. Numer. Sinica 11 (1982) 445–450.
[8] C. Dong, A family of multipoint iterative functions for finding multiple roots of equations, Intern. J. Comput. Math. 21 (1987) 363–367.
[9] E. Hansen, M. Patrick, A family of root finding methods, Numer. Math. 27 (1977) 257–269.
[10] H.D. Victory, B. Neta, A higher order method for multiple zeros of nonlinear functions, Int. J. Comput. Math. 12 (1983) 329–335.
[11] N. Osada, An optimal multiple root-finding method of order three, J. Comput. Appl. Math. 51 (1994) 131–133.
[12] B. Neta, New third order nonlinear solvers for multiple roots, Appl. Math. Comput. 202 (2008) 162–170.
[13] D. Sbibih, A. Serghini, A. Tijini, A. Zidna, A general family of third order method for finding multiple roots, Appl. Math. Comput. 233 (2014) 338–350.
[14] C. Chun, B. Neta, A third-order modification of Newton's method for multiple roots, Appl. Math. Comput. 211 (2009) 474–479.
[15] H.H.H. Homeier, On Newton-type methods for multiple roots with cubic convergence, J. Comput. Appl. Math. 231 (2009) 249–254.
[16] Y.H. Geum, Y.I. Kim, Cubic convergence of parameter-controlled Newton-secant method for multiple zeros, J. Comput. Appl. Math. 233 (2009) 931–937.
[17] Y.I. Kim, Y.H. Geum, A cubic-order variant of Newton's method for finding multiple roots of nonlinear equations, Comput. Math. Appl. 62 (2011) 249–254.
[18] B. Neta, C. Chun, On a family of Laguerre methods to find multiple roots of nonlinear equations, Appl. Math. Comput. 219 (2013) 10987–11004.
[19] B. Neta, A.N. Johnson, High order nonlinear solver for multiple roots, Comput. Math. Appl. 55 (2008) 2012–2017.
[20] B. Neta, Extension of Murakami's high order nonlinear solver to multiple roots, Intern. J. Comput. Math. 8 (2010) 1023–1031.
[21] S.G. Li, L.Z. Cheng, B. Neta, Some fourth-order nonlinear solvers with closed formulae for multiple roots, Comput. Math. Appl. 59 (2010) 126–135.
[22] B. Liu, X. Zhou, A new family of fourth-order methods for multiple roots of nonlinear equations, Nonlinear Anal. Model. Control 18 (2) (2013) 143–152.
[23] V. Kanwar, S. Bhatia, M. Kansal, New optimal class of higher-order methods for multiple roots, permitting $f'(x_n) = 0$, Appl. Math. Comput. 222 (2013) 564–574.
[24] X. Zhou, X. Chen, Y. Song, Construction of higher order methods for multiple roots of nonlinear equations, J. Comput. Appl. Math. 235 (2011) 4199–4206.
[25] H.T. Kung, J.F. Traub, Optimal order of one-point and multipoint iterations, J. Assoc. Comput. Mach. 21 (1974) 643–651.
[26] C. Chun, B. Neta, Basins of attraction for Zhou-Chen-Song fourth order family of methods for multiple roots, Math. Comput. Simul. 109 (2015) 74–91.
[27] B.D. Stewart, Attractor basins of various root-finding methods (M.S. thesis), Naval Postgraduate School, Department of Applied Mathematics, Monterey, CA, June 2001.
[28] S. Amat, S. Busquier, S. Plaza, Iterative root-finding methods, unpublished report (2004).
[29] S. Amat, S. Busquier, S. Plaza, Review of some iterative root-finding methods from a dynamical point of view, Scientia 10 (2004) 3–35.
[30] S. Amat, S. Busquier, S. Plaza, Dynamics of a family of third-order iterative methods that do not require using second derivatives, Appl. Math. Comput. 154 (2004) 735–746.
[31] S. Amat, S. Busquier, S. Plaza, Dynamics of the King and Jarratt iterations, Aeq. Math. 69 (2005) 212–2236.
[32] M. Scott, B. Neta, C. Chun, Basin attractors for various methods, Appl. Math. Comput. 218 (2011) 2584–2599.
[33] C. Chun, M.Y. Lee, B. Neta, J. Džunić, On optimal fourth-order iterative methods free from second derivative and their dynamics, Appl. Math. Comput. 218 (2012) 6427–6438.
[34] A.A. Magreñán, Different anomalies in a Jarratt family of iterative root-finding methods, Appl. Math. Comput. 233 (2014) 29–38.
[35] I.K. Argyros, A.A. Magreñán, On the convergence of an optimal fourth-order family of methods and its dynamics, Appl. Math. Comput. 252 (2015) 336–346.

[36] A. Cordero, L. Feng, A.A. Magreñán, J.R. Torregrosa, A new fourth-order family for solving nonlinear problems and its dynamics, J. Math. Chem. 53 (2015) 893–910.
[37] F. Chicharro, A. Cordero, J.M. Gutiérrez, J.R. Torregrosa, Complex dynamics of derivative-free methods for nonlinear equations, Appl. Math. Comput. 219 (2013) 7023–7035.
[38] A. Cordero, J. García-Maimó, J.R. Torregrosa, M.P. Vassileva, P. Vindel, Chaos in King's iterative family, Appl. Math. Lett. 26 (2013) 842–848.
[39] B. Neta, M. Scott, C. Chun, Basin of attractions for several methods to find simple roots of nonlinear equations, Appl. Math. Comput. 218 (2012) 10548–10556.
[40] C. Chun, B. Neta, S. Kim, On Jarratt's family of optimal fourth-order iterative methods and their dynamics, Fractals 22 (2014), doi:10.1142/S0218348X14500133.
[41] B. Neta, M. Scott, C. Chun, Basin attractors for various methods for multiple roots, Appl. Math. Comput. 218 (2012) 5043–5066.
[42] B. Neta, C. Chun, Basins of attraction for several optimal fourth order methods for multiple roots, Math. Comput. Simul. 103 (2014) 39–59.
[43] E.R. Vrscay, W.J. Gilbert, Extraneous fixed points, basin boundaries and chaotic dynamics for Schröder and König rational iteration functions, Numer. Math. 52 (1988) 1–16.